



Scholars' Mine

Masters Theses

Student Theses and Dissertations

1970

An approximate min-max solution to the identification problem

Richard Loyd Warren

Follow this and additional works at: https://scholarsmine.mst.edu/masters_theses

 Part of the [Electrical and Computer Engineering Commons](#)

Department:

Recommended Citation

Warren, Richard Loyd, "An approximate min-max solution to the identification problem" (1970). *Masters Theses*. 5405.

https://scholarsmine.mst.edu/masters_theses/5405

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

AN APPROXIMATE MIN-MAX SOLUTION
TO THE IDENTIFICATION PROBLEM

BY
RICHARD LOYD WARREN, 1943-

A
THESIS
submitted to the faculty of
UNIVERSITY OF MISSOURI - ROLLA
in partial fulfillment of the requirements for the
Degree of
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
Rolla, Missouri
1970

Approved by

Herbert A. Cusby (advisor)

Richard T. Johnson

John L. Payne

ABSTRACT

A technique based on the min-max error criterion is proposed for identifying a dynamic system in terms of a discrete-time model from the system response to a deterministic input. A linear, single-input/single-output, lumped-parameter, time-invariant (at least during the measurement interval) system is assumed. Although the resulting problem is non-linear, it is shown that linear programming techniques are applicable if a realistic approximation is made.

The technique is implemented on the digital computer and evaluated by considering a number of typical discrete-time systems. The effects of additive noise are also considered.

ACKNOWLEDGEMENTS

The author wishes to thank Dr. Herbert Crosby for his suggestions leading to the formulization of the algorithm presented in this thesis. He is also grateful for the use of the McDonnell Douglas digital computing facilities.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
LIST OF ILLUSTRATIONS	v
I. INTRODUCTION	1
A. ADAPTIVE CONTROL	1
B. DIGITAL SIMULATION	2
C. THE IDENTIFICATION PROBLEM	4
II. REVIEW OF LITERATURE	6
III. DISCUSSION	8
A. MIN-MAX CURVE FITTING	8
B. RESPONSE OF DISCRETE-TIME SYSTEMS	11
C. THE IDENTIFICATION TECHNIQUE	16
D. APPLICATIONS AND EVALUATION	19
IV. CONCLUSION	30
V. APPENDICES	31
A. THE SIMPLEX ALGORITHM	31
B. COMPUTER PROGRAM DESCRIPTION	36
C. LISTING OF FORTRAN IV COMPUTER PROGRAM IDNT ..	49
VI. BIBLIOGRAPHY	59
VII. VITA	63

LIST OF ILLUSTRATIONS

Figures	Page
1. Effect of Additive Noise on Parameter Estimate Errors	25
2. Effect of Additive Noise on Parameter Identification With Delay	26
3. Effect of Additive Noise on Parameter Identification Without Delay	27
A1. The Two Dimensional Linear Programming Problem	33
B1. Executive Program IDNT Flow Chart	37
B2. Flow Chart of Matrix Generating Subroutine MTRX ...	40
B3. Flow Chart of Simplex Algorithm Subroutine SMPLX ..	43
B4. Flow Chart of Sorting Subroutine SORT	46
B5. Flow Chart of Z-Transform Inversion Subroutine FCAL	48

I. INTRODUCTION

The fundamental problem of identifying the dynamic characteristics of systems from observations of input and output functions over a finite interval of time is important in a number of diversified fields. Accurate determination of system characteristics on a real-time scale is particularly important in adaptive control system techniques, in which periodic controller "redesign" maintains consistent system performance regardless of plant variations. An analogous problem occurs in communication systems involving optimum utilization of time-varying channels as described by Price and Green (1958). In the study of complex systems, where an analytical derivation of the defining differential equations is not practical, identification techniques afford a workable solution. Of particular importance, in this respect, is the recent use of these techniques in developing models of various biological functions, including human operators, as described in McRuer et al. (1965). A distinctly different application is the generation of discrete models of continuous-time systems for digital computer simulation. Of these applications, the first and last are perhaps most important and will be considered in some detail.

A. ADAPTIVE CONTROL

Modern adaptive control techniques have evolved as a result of the requirement that non-stationary systems,

such as high-performance aircraft and sophisticated chemical processes, be optimally controlled (in some sense) despite their parameter variations. If the open-loop plant transfer function is available, any one of a number of controller design techniques based on a wide range of optimal control schemes can be applied to realize the desired performance. The techniques proposed by Bertram (1956), Anderson et al. (1958), Cosgriff and Emerling (1958), Eykhoff (1960), Desoer and Wing (1961), and Zaborsky and Berger (1962) are typical of the applicable controller design schemes. The major problem, thus, is to determine the system parameters continually or at a rate much faster than the rate of parameter variations. This implies that the parameter variations must be slow with respect to the dominant system time constants in order that a transfer function representation is meaningful. Obviously, the presence of measurement noise precludes an exact identification in a finite time interval, and thus, a trade-off between accuracy (noise filtering) and identification time is required.

B. DIGITAL SIMULATION

Digital computer simulation of all types of dynamic physical systems has increased considerably in recent years with the significant improvements in computer speed and methodology. The digital computer has practically replaced the analog computer for real-time flight simulation and

"quick-look" simulations of first-cut designs. The fundamental problem in digital simulation of such systems is, of course, the transformation of continuous-time dynamic characteristics, represented either empirically or analytically, into a discrete-time model amenable to digital processing.

Although there are a number of very useful simulation languages presently available which perform this task for an analytical model via macro-instruction techniques, the resulting running times, in most cases, are prohibitively long for essentially real-time calculations. The techniques of Tustin (1947), Truxal (1954), Boxer-Thaler (1956), and Sage-Smith (1966) can be used to generate a digital model directly from a transfer function representation of the system. When the system is represented by a set of first-order differential equations, numerical integration methods, such as Runge-Kutta-Blum (1952) or Adams-Moulton (see Hamming (1962)), may be applied.

A distinctly different problem occurs when the system to be simulated is defined only in terms of input/output data. Identification of the system in terms of the parameters of an appropriate discrete model defines the digital simulation procedure. The particular identification scheme to be used depends to a large extent on whether the input can be best described analytically or statistically. Existing techniques applicable in either case are outlined

in the Review of Literature.

C. THE IDENTIFICATION PROBLEM

The objective of any system identification technique is the estimation of a set of parameters which adequately describes the system transfer characteristics. In general, these transfer characteristics should enable the output of the system to be calculated in response to any input or control signal. Based on this premise, the fundamental factors to be considered are:

1. Form of the model,
2. Sensitivity to stored energy,
3. Amount of a priori information,
4. Practicality of special test signals, and
5. Identification time and accuracy.

Obviously, the structure of the model must be limited to some extent to make the problem meaningful. The amount of a priori information available is a prime factor in the formulation of that model and determines, to a large extent, the structure of the identification technique itself. The other factors are determined principally by whether the identification in a particular application is basically real-time or an after-the-fact calculation. For example, a technique which requires special test signals and a zero energy state would be impractical in most adaptive control systems but function favorably in the identification of a human operator model. Thus, in

general, each requirement must be determined from a detailed analysis of the specific application.

The objective of this thesis is to develop a broadly applicable identification technique based on the min-max error criterion and requiring a minimum of a priori information. The identification is based on input/output data of a deterministic nature and a model constrained as follows:

1. Linear,
2. Discrete,
3. Lumped-parameter,
4. Single input/single output,
5. Stationary, and
6. Low-pass.

The outputs of the identification procedure are estimates of the parameters (a_j, b_j) of the discrete transfer function

$$D(z) = \frac{a_0 + a_1 z^{-1} + \dots + a_p z^{-p}}{1 + b_1 z^{-1} + \dots + b_q z^{-q}}, \quad (1)$$

where the order of the denominator is assumed equal to the order of the numerator ($p = q$).

II. REVIEW OF LITERATURE

Due to its importance in the fields of adaptive control techniques and digital simulation, the real-time system identification problem has received considerable attention in the literature, particularly from 1958 to 1967. With few exceptions, the techniques developed assumed a linear (discrete or differential) model for a time-varying, quasi-linear system. The first significant article addressing the general problem was presented by Kalman (1958). The technique he proposed is based on minimization of the weighted-mean-square error for a pulse transfer function model and is directed toward design of a self-optimizing controller. Earlier, Goodman and Reswick (1955), Margolis (1955), and Weisner and Lee (1950) presented techniques based on cross-correlation of input and output signals which are essentially limited to time-invariant systems if a complete identification is desired. Levin (1960) applied statistical estimation theory to derive a method optimal in the least squares sense when sample values are contaminated with noise. The derivation of Kerr and Surber (1961) provides bounds on the integral-squared error for identification with various input signals. Joseph, Lewis, and Tou (1961) investigated the performance of digital adaptive systems incorporating their identification algorithm with promising results. A rather novel scheme involving integrals of both input and

output signals was suggested by Zaborsky and Berger (1962) and applied to self-optimizing control based on the integral-square-of-error-by-time performance criterion. The identification procedure described by Kushner (1962) is a first order iterative process equivalent to the steepest decent method. Estimation of the coefficients in the system pulse transfer function in a generalized least squares sense is the object of the work of Levin (1964). The problem of nonlinear sampled-data system identification was addressed by Steiglitz and McBride (1965). Recently, Liapunov design techniques were applied to the identification problem by Pazdera and Pottinger (1969).

A multitude of variations on these identification techniques have been proposed and a substantial listing of publications is presented in the Bibliography.

III. DISCUSSION

A. MIN-MAX CURVE FITTING

All time domain identification techniques reduce to the fundamental curve fitting problem. A general statement of this problem for the discrete case is: determine α_j , $j = 1, \dots, n$, such that a norm of the error $\delta(\alpha, k)$,

$$\delta(\alpha, k) = w(k) \left[f(k) - \sum_{j=1}^n \alpha_j g_j(k) \right], \quad (2)$$

is minimized over the interval $k = 0, \dots, s$, where $f(k)$ is the function (or sequence) to be approximated, $w(k)$ is a weighting function, and $\{g_j(k), j = 1, \dots, n\}$ is a set of suitable functions. Most norms of practical interest are of the form

$$L_p(\alpha) = \left[\sum_{k=1}^s |\delta(\alpha, k)|^p \right]^{1/p}, \quad (3)$$

where p is a positive integer.

An identification scheme is defined by specification of:

1. The set of functions $\{g_j(k), j = 1, \dots, n\}$ (or equivalently, the form of the system description desired),
2. The weighting function $w(k)$,
3. The norm to be used, and
4. The minimization technique.

Obviously, the variations are innumerable. For the mathematically tractable case with $p = 1$, which is the "least-squares" criterion, numerous solutions have been proposed as described in the Review of Literature. In general, these solutions result in linear systems of equations with summed cross-products of input and output samples as coefficients. Steepest descent and random search techniques have also been applied to the minimization of this and other norms. Appropriate weighting functions are employed in certain applications to emphasize the most recent samples, larger errors, etc.

Minimization of the norm defined in Equation 3 for $p = \infty$ is equivalent to minimizing the functional

$$\lambda(\alpha) = \max_k \left| \delta(\alpha, k) \right|, \quad (4)$$

or more simply, minimizing the maximum error. Although this min-max concept is intuitively satisfying, applications to curve fitting problems are rarely seen. In the following, a general min-max solution based on linear programming techniques is developed for the curve fitting problem and applied to the identification of discrete systems.

A somewhat more manageable statement of the min-max criterion for this application is: determine α_j , $j = 1, \dots, n$ and the minimum λ satisfying

$$\left| f(k) - \sum_{j=1}^n \alpha_j g_j(k) \right| \leq \lambda, \quad (5)$$

for $k = 0, \dots, s$. The absolute value may be eliminated by writing Equation 5 as two equations,

$$\lambda + \sum_{j=1}^n \alpha_j g_j(k) \geq f(k) \quad (6a)$$

$$-\lambda + \sum_{j=1}^n \alpha_j g_j(k) \leq f(k), \quad (6b)$$

for $k = 0, \dots, s$. Comparison of this statement of the min-max criterion with the general linear programming problem described in Appendix A reveals that a solution may be obtained through application of the simplex algorithm with the objective function

$$Z = -\lambda. \quad (7)$$

At this point, the reader unfamiliar with linear programming techniques should refer to Appendix A for a brief summary of the notation and theory of the simplex algorithm. In general, the algorithm provides an iterative, computationally efficient solution to the minimization (or maximization) problem involving linear constraints and a linear objective function. Although the basic algorithm requires that the right-hand-side of Equation 6

and all the variables be non-negative, this restriction may be sidestepped by appropriate manipulations which will be described shortly.

B. RESPONSE OF DISCRETE-TIME SYSTEMS

A linear, discrete-time system may be represented analytically by any of the following:

1. Discrete transfer function (z-transform)

$$D(z) = \frac{a_0 + a_1 z^{-1} + \dots + a_p z^{-p}}{1 + b_1 z^{-1} + \dots + b_q z^{-q}} \quad (8)$$

2. Weighting sequence (impulse response)

$$d(k), \quad k = 0, 1, 2, \dots, \infty,$$

with

$$c(k) = \sum_{j=1}^k d(j)r(k-j), \quad (9)$$

3. Difference equation

$$b_q c(k-q) + b_{q-1} c(k-q-1) + \dots + b_1 c(k-1) + \quad (10)$$

$$c(k) = a_p r(k-p) + \dots + a_1 r(k-1) + a_0 r(k),$$

4. State variable model

$$\begin{aligned} \overline{x(k+1)} &= \overline{A} \overline{x(k)} + \overline{B} \overline{r(k)} \\ \overline{c(k)} &= \overline{D} \overline{x(k)} + \overline{E} \overline{r(k)}, \end{aligned} \quad (11)$$

where capital letters represent matrices and lower case represent vectors.

In each of the above, k is the independent variable, and R and C denote the system input and output, respectively.

Although each of these models may be used to determine the system output for a given input, they are quite different in many respects and individually provide a variety of information on system operation. The discrete transfer function in the z -domain ($z = e^{+st}$, s is the Laplace variable) yields much the same information as does the s -domain transfer function for continuous systems. Stability considerations are generally based on analysis of the discrete transfer function or the state variable model. The system weighting sequence relates directly to the transient response and may be used to calculate the output sequence by convolution with the input. The difference equation representation is used extensively in digital simulation. Finally, the state variable model is the basis for discrete optimal and stochastic control theory.

For the purposes of system identification from empirical data, the discrete transfer function is perhaps the most efficient representation for most applications. This is due primarily to the relatively small number of parameters required and the prominent use of these parameters in optimal controller design, digital simulation, and filter design. Also, transformation of the transfer function representation into any of the other models is reasonably straightforward. Based on these considerations, an identification model of the form of Equation 8 is assumed.

The discrete transfer function $D(z)$ is uniquely related to the system impulse (or Kronecker delta) response by the z -transform,

$$D(z) = \sum_{n=0}^{\infty} d(n)z^{-n}. \quad (12)$$

For the linear, stationary, discrete-time system, $D(z)$ may be written as the ratio of two polynomials in z^{-k} as in Equation 8. The inverse z -transform of $D(z)$ is given by

$$d(k) = \frac{1}{2\pi j} \oint D(z) z^{k-1} dz \quad (13)$$

or by synthetic division of the right-hand-side of Equation 8. Freeman (1965, pp. 213-215) has shown that, for the latter case, the result may be written

$$d(k) = a_k - \sum_{j=1}^q b_j d(k-j), \quad k = 0, \dots, n, \quad (14)$$

where q = order of the denominator of $D(z)$, and

p = order of the numerator of $D(z)$.

If $k < q$, the upper limit on the summation becomes k , and $a_k = 0$ for $k > p$. The parameter b_0 has been set equal to one, as in Equation 8, to normalize the model. This in no way affects the generality of the model.

Both the system impulse response of infinite length and the discrete transfer function are unique representations of a discrete-time system. Furthermore, these two

models are uniquely related as given in Equations 12 and 13. Given a set of transfer function coefficients, the impulse response may be easily calculated from Equation 14. There is, however, no known exact method for obtaining the discrete-time transfer function coefficients from the impulse response. The difficulty arises in obtaining a closed-form expression for an infinite sequence.

For the case of a truncated impulse response (a finite sequence), uniqueness between the transfer function representation and the impulse response is not assured. In fact, there exists an infinite number of transfer functions which invert by Equation 14 to yield the sequence $d(k)$, $k = 0, \dots, s$, where s is finite. For example, consider the expansion of Equation 14 for $s+1$ points.

$$\begin{aligned}
 d(0) &= a_0 \\
 d(1) &= a_1 - b_1 d(0) \\
 d(2) &= a_2 - b_2 d(0) - b_1 d(1) \\
 &\vdots \\
 d(s) &= a_s - b_s d(0) - b_{s-1} d(1) - \dots - b_1 d(s-1).
 \end{aligned} \tag{15}$$

With $s+1$ equations and $2s+1$ unknown parameters, the set of non-linear simultaneous equations has many solutions, and thus, the parameters cannot be determined.

If the sequence at the input to the identification scheme represents a system response to a sequence other than an impulse, the impulse response may be calculated

from

$$C(z) = D(z)R(z), \quad (16)$$

where $R(z)$ is the z -transform of the input sequence.

The impulse response is thus

$$D(z) = \frac{C(z)}{R(z)}. \quad (17)$$

For example, if the input is a step,

$$R(z) = \frac{z}{z - 1}, \quad (18)$$

the impulse response is given by

$$D(z) = [1 - z^{-1}]C(z). \quad (19)$$

If $D(z)$ and $C(z)$ are written

$$\begin{aligned} C(z) &= c_0 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_n z^{-n} + \dots \\ D(z) &= d_0 + d_1 z^{-1} + d_2 z^{-2} + \dots + d_n z^{-n} + \dots \end{aligned} \quad (20)$$

Substitution in Equation 17 gives

$$\begin{aligned} d_0 + d_1 z^{-1} + d_2 z^{-2} + \dots + d_n z^{-n} + \dots = \\ [1 - z^{-1}][c_0 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_n z^{-n} + \dots]. \end{aligned} \quad (21)$$

Equating coefficients of like powers of z yields the result

$$\begin{aligned} d_0 &= c_0 \\ d_1 &= c_1 - c_0 \\ d_2 &= c_2 - c_1 \\ &\vdots \\ &\vdots \end{aligned} \quad (22)$$

$$d_n = c_n - c_{n-1},$$

which is the impulse response sequence.

For the case of a sine input,

$$R(z) = \frac{z \sin \omega t}{z^2 - 2z \cos \omega t + 1} \cdot \quad (23)$$

Proceeding as before,

$$\begin{aligned} d_0 &= (c_1 - 2\cos(\omega)c_0)/\sin(\omega) \\ d_1 &= (c_2 - 2\cos(\omega)c_1 + c_0)/\sin(\omega) \\ &\vdots \\ d_n &= (c_{n+1} - 2\cos(\omega)c_n + c_{n-1})/\sin(\omega). \end{aligned} \quad (24)$$

Similar relationships for other input sequences may be derived by the same technique.

C. THE IDENTIFICATION TECHNIQUE

Based on the analysis in the previous sections, a min-max formulation of the time-domain identification problem may be obtained by substitution of Equation 14 into Equation 6.

$$\lambda + \tilde{a}_k - \sum_{j=1}^q \tilde{b}_j \tilde{d}(k-j) \geq d(k) \quad (25a)$$

$$-\lambda + \tilde{a}_k - \sum_{j=1}^q \tilde{b}_j \tilde{d}(k-j) \leq d(k) \quad (25b)$$

with

$$Z = -\lambda \quad (25c)$$

for $k = 0, 1, \dots, s$, where \tilde{a} indicates an estimated value and the other variables are as defined earlier. If $k < q$, the upper limit on the summations in Equation 25 is k , and $\tilde{a}_k = 0$ for $k > p$.

Equation 25 is obviously non-linear since the $\tilde{d}(k-j)$ terms are functions of the parameter estimates. For example, for the case $p = q = 1$

$$\begin{aligned} \tilde{d}(0) &= \tilde{a}_0 \\ \tilde{d}(1) &= \tilde{a}_1 - \tilde{b}_1 \tilde{d}(0) \\ \tilde{d}(2) &= -\tilde{b}_1 \tilde{d}(1) \\ \tilde{d}(3) &= -\tilde{b}_1 \tilde{d}(2) \\ &\vdots \\ \tilde{d}(k) &= -\tilde{b}_1 \tilde{d}(k-1), \end{aligned} \quad (26)$$

and thus,

$$\begin{aligned} \tilde{d}(0) &= \tilde{a}_0 \\ \tilde{d}(1) &= \tilde{a}_1 - \tilde{b}_1 \tilde{a}_0 \\ \tilde{d}(2) &= -\tilde{b}_1 \tilde{a}_1 + \tilde{b}_1^2 \tilde{a}_0 \\ \tilde{d}(3) &= \tilde{b}_1^2 \tilde{a}_1 - \tilde{b}_1^3 \tilde{a}_0 \\ &\vdots \\ \tilde{d}(k) &= (-1)^{k-1} \tilde{b}_1^{k-1} \tilde{a}_1 + (-1)^k \tilde{b}_1^k \tilde{a}_0. \end{aligned} \quad (27)$$

Although minimization techniques, such as gradient methods or random search may be applied to obtain a solution of the problem in this form, these techniques are inefficient and cumbersome, especially if higher-order systems are considered. Furthermore, these techniques provide only an

approximate solution to the problem as stated.

Examination of Equation 25 indicates that if the approximation

$$\tilde{d}(k) \approx d(k) \quad (28)$$

is made, the problem reduces to the linear programming problem discussed earlier. Substitution into Equation 25 yields

$$\lambda + \tilde{a}_k - \sum_{j=1}^q \tilde{b}_j d(k-j) \geq d(k) \quad (29a)$$

$$-\lambda + \tilde{a}_k - \sum_{j=1}^q \tilde{b}_j d(k-j) \leq d(k) \quad (29b)$$

with

$$Z = -\lambda \quad (29c)$$

under the same conditions as Equation 25. Such an approximation results in the following:

1. The value of λ in the solution of Equation 29 is not exactly the min-max error, and
2. Convergence of the parameter estimates is not assured.

The resulting identification scheme, however, does benefit from the attractive features of the simplex algorithm, such as computational efficiency and rapid convergence to a solution.

The statement of the problem as given in Equation 29 does not provide for negative values of the parameter estimates and the impulse response samples when the

solution is obtained by the simplex algorithm. Negative values of the parameters may be allowed by making the substitutions,

$$\begin{aligned} a_k &= a_k^+ - a_k^-, \\ b_j &= b_j^+ - b_j^-, \end{aligned} \tag{30}$$

in Equation 29 with all the new variables constrained to be non-negative. For a negative $d(k)$, multiplication of Equations 29a and 29b by minus one is all that is required to obtain the proper form.

The one remaining difficulty is: with no a priori information, how is the order of the system (p and q in Equation 29) to be determined? The identification scheme as presented provides no basis for this determination. It is necessary, therefore, to employ a trial and error approach. One such approach is: starting with an initial estimate, the identification procedure is applied and the order increased iteratively until a preset bound on the min-max error is satisfied. Although this scheme is crude and simple, the results of the next section testify to its effectiveness. For $p = q$ and an error bound approximately two-orders-of-magnitude greater than the round-off error of the computer, the scheme did not fail once.

D. APPLICATIONS AND EVALUATION

In order to determine and demonstrate the performance

of the identification technique, a series of examples are considered which effectively test the method for the more difficult aspects of the identification problem. Specific areas that are investigated include order determination, pure delays, effects of the approximation, biases in the samples, and noise sensitivity.

A digital computer implementation of the technique, which is presented in Appendices B and C, was employed in the evaluation. Other subroutines were developed to provide inputs where necessary. A Control Data 6400 digital computer (approximately 15-place arithmetic) was used for all computations.

Example 1. Sensitivity to Correct System Order

The first example was selected to evaluate the capability of the proposed technique to determine the order of discrete-time systems as part of the identification process. A number of situations were considered in evaluating this important aspect of the identification technique.

For the discrete transfer function with poles of the same order-of-magnitude,

$$D(z) = \frac{z^{-m}}{(1 - 0.9z^{-1})(1 - 0.8z^{-1})(1 - 0.7z^{-1})} \quad (31)$$

$$= \frac{z^{-m}}{1 - 2.4z^{-1} + 1.91z^{-2} - 0.504z^{-3}}, \quad (32)$$

the impulse response, as given by Equation 14, was calculated for $k = 1, \dots, 20$. Application of the identification technique with p -initial equal to 1 and an order determination error bound of 10^{-8} , produced the following results for $m = 0, \dots, 4$:

1. The iteration terminated at $p = q = 3$ for all m ,
2. The denominator parameters were determined to better than 13 significant figures,
3. The numerator parameters not equal to 1.0 were also on the order of 10^{-13} ,
4. The value of λ was approximately 10^{-13} ,
5. The actual maximum error was approximately 10^{-10} ,
and

6. Zeros in the response did not affect the solution.

For the system with poles differing by orders-of-magnitude,

$$D(z) = \frac{z^{-m}}{(1 - 0.9z^{-1})(1 - 0.09z^{-1})(1 - 0.009z^{-1})} \quad (33)$$

$$D(z) = \frac{z^{-m}}{1 - 0.999z^{-1} + 0.8991z^{-2} - 0.000729z^{-3}}, \quad (34)$$

similar results were obtained. In both cases, the accuracy of the solution is bounded by the propagation of round-off errors in the calculations.

Example 2. High-Order Systems

An identification based on 20 samples of the impulse

response for

$$D(z) = \frac{(1 - 0.2z^{-1})(1 - 0.3z^{-1})z^{-1}}{(1 - 0.9z^{-1})(1 - 0.09z^{-1})(1 - 0.009z^{-1})} \quad (35)$$

yielded the results:

$$\begin{aligned} p &= q = 3 \\ a_0 &= -1.421 \times 10^{-14} \\ a_1 &= 1.000000000000178 \\ a_2 &= -0.5000000000002578 \\ a_3 &= 0.0600000000000921 \\ b_1 &= -0.9990000000000015 \\ b_2 &= 0.0899100000000355 \\ b_3 &= -0.0007290000000113 \\ \lambda &= 1.421 \times 10^{-14} \\ L &= 1.172 \times 10^{-13} \end{aligned}$$

where L is the actual maximum error (measured). Likewise, for

$$D(z) = \frac{(1 - 0.5z^{-1} + 0.06z^{-2})z^{-1}}{1 - 2.4z^{-1} + 1.91z^{-2} - 0.504z^{-3}} \quad (36)$$

the results are:

$$\begin{aligned} a_0 &= -8.701 \times 10^{-14} \\ a_1 &= 1.000000000000648 \\ a_2 &= -0.500000000000161 \\ a_3 &= 0.0600000000001342 \\ b_1 &= -2.4000000000000067 \end{aligned}$$

$$\begin{aligned}
b_2 &= 1.910000000000122 \\
b_3 &= -0.504000000000293 \\
\lambda &= 8.701 \times 10^{-14} \\
L &= 9.398 \times 10^{-11} \\
p &= q = 3
\end{aligned}$$

Example 3. Repeated Roots

From 20 samples of the impulse response of the system

$$D(z) = \frac{z^{-1}}{(1 - 0.9z^{-1})^2}, \quad (37)$$

$$= \frac{z^{-1}}{1 - 1.8z^{-1} + 0.81z^{-2}}, \quad (38)$$

the identification scheme calculated:

$$\begin{aligned}
p &= q = 2 \\
a_0 &= -2.309 \times 10^{-14} \\
a_1 &= 1.000000000000788 \\
a_2 &= -4.504 \times 10^{-14} \\
b_1 &= -1.800000000000087 \\
b_2 &= 0.810000000000591 \\
\lambda &= 2.309 \times 10^{-14} \\
L &= 1.174 \times 10^{-11}
\end{aligned}$$

Example 4. Noise Sensitivity

The effect of additive (measurement) noise on the estimation of parameters for the system,

$$D(z) = \frac{z^{-1}}{1 - 0.9z^{-1}}, \quad (39)$$

is presented in Figure 1. For each parameter, E is the mean-absolute-error,

$$E = \frac{1}{N} \sum_{i=1}^N |e_i|, \quad (40)$$

where e is the error associated with a single measurement. The gaussian noise samples added to the impulse response samples are characterized by their standard deviation σ . Twelve samples were used in the identification and $N = 50$.

Example 5. Redundancy Factor

The number of response samples required for the identification of a system is ideally

$$n_i = p + q - 1. \quad (41)$$

If noise is present, however, the results of an identification based on that number of samples would most likely be inaccurate. Accuracy is generally improved by taking an average over more samples. Figure 2 illustrates the effects of increasing the number of samples used in the identification of the system of Equation 39, where

$$\text{Redundancy Factor} = \frac{\text{No. of Samples}}{n_i}. \quad (42)$$

Figure 3 provides the same information for a similar system with no delay ($a_0 = 1$, $a_1 = 0$). In both cases, the

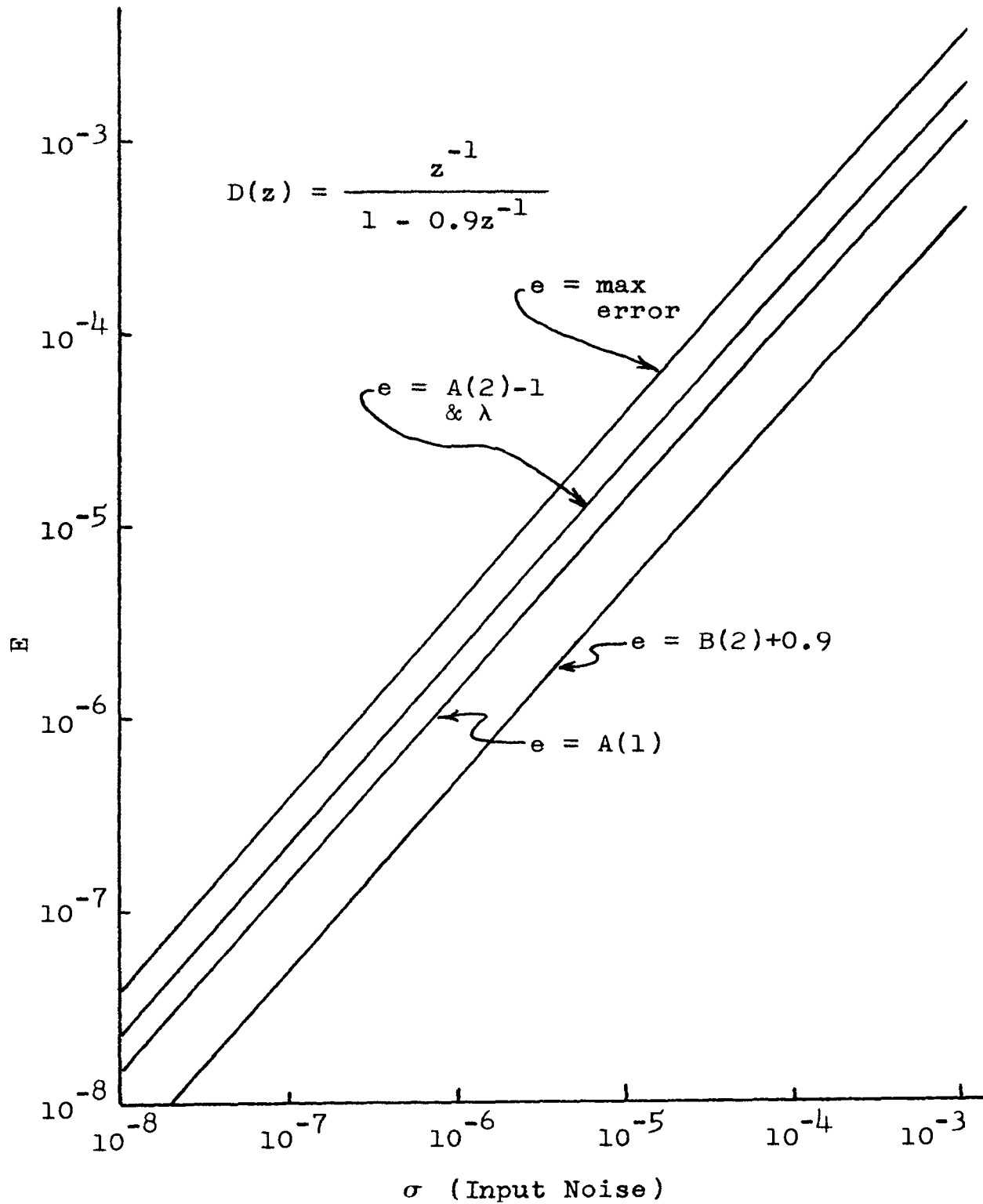


Fig. 1. Effect of Additive Noise Level
on Parameter Estimate Errors.

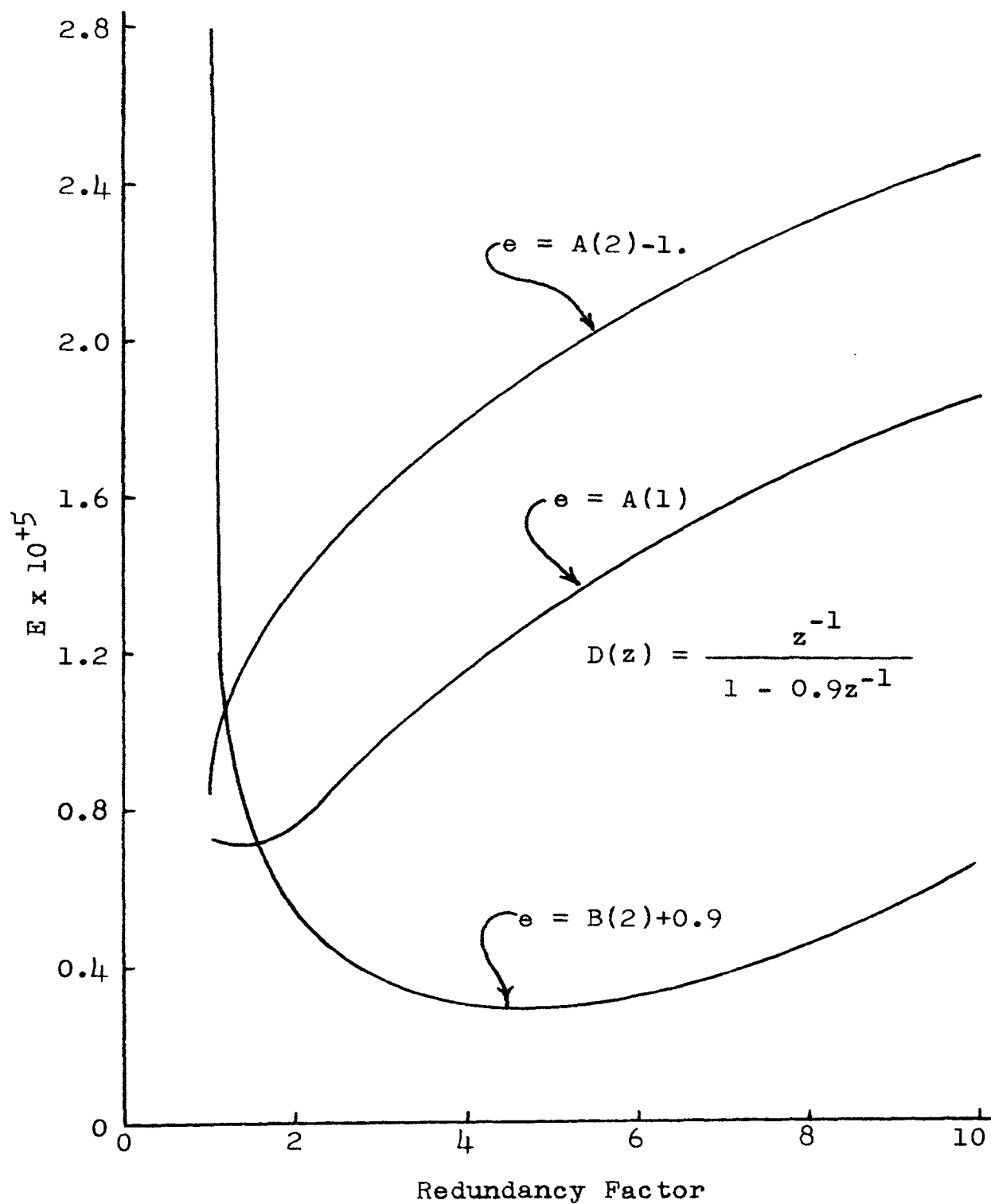


Fig. 2. Effect of Additive Noise on Parameter Identification With Delay.

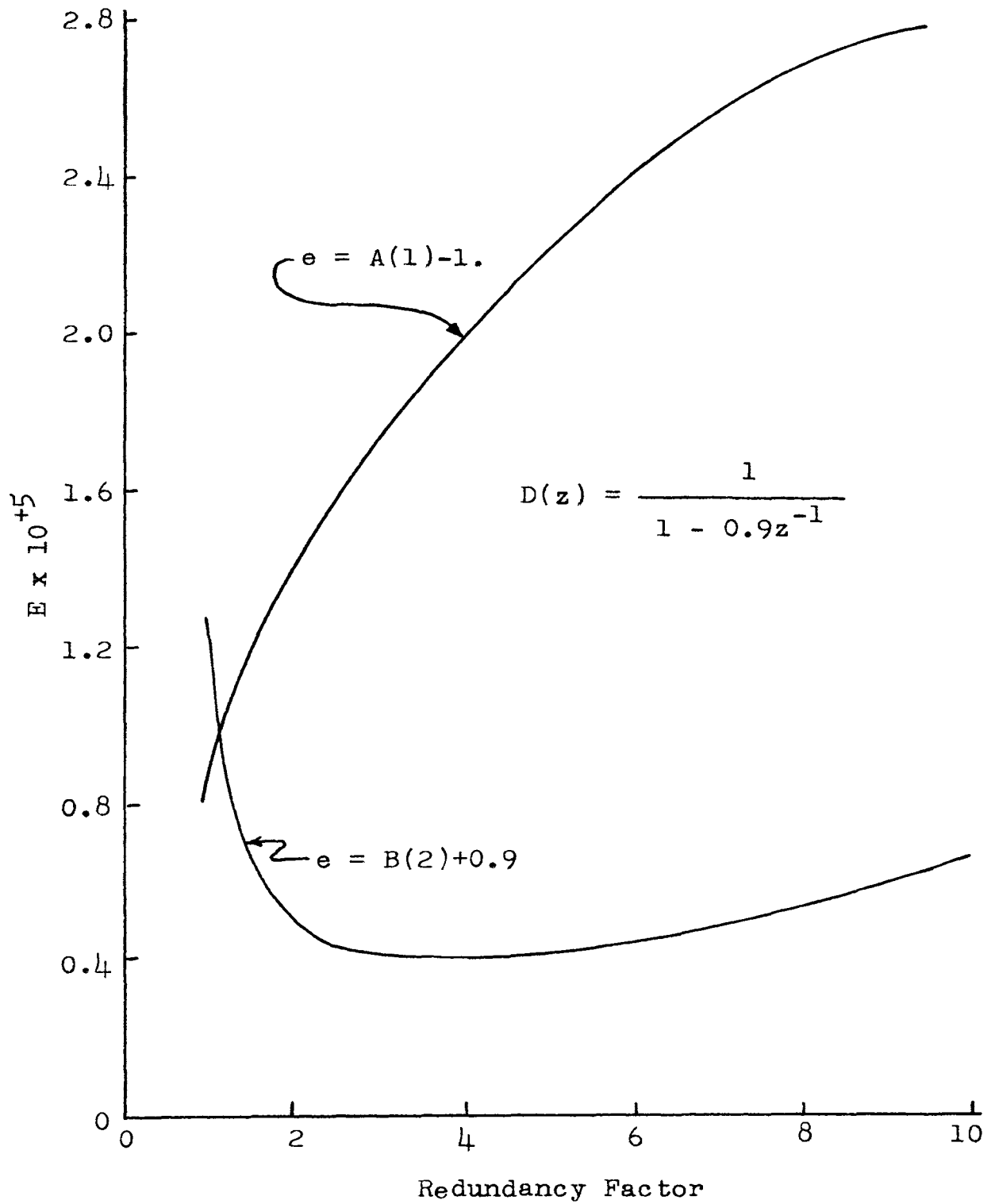


Fig. 3. Effect of Additive Noise on Parameter Identification Without Delay.

standard deviation of the additive noise was .00001.

Example 6. Identification With Sinusoid Input

The response of the system of Equation 39 to the input,

$$r(n) = \sin(0.05n), \quad (43)$$

was used for identification by calculating the impulse response by the method discussed in Response of Discrete-Time Systems (the subroutine of Appendix C). The results are:

$$\begin{aligned} p &= q = 1 \\ a_0 &= -1.048 \times 10^{-12} \\ a_1 &= 1.000000000000747 \\ b_1 &= -0.900000000000312 \\ \lambda &= 1.048 \times 10^{-12} \\ L &= 1.048 \times 10^{-12} \end{aligned}$$

Twenty samples were used in the identification.

Example 7. Biased Samples

If a bias is present in the output of the system of Equation 39, the measured impulse response correspondes to

$$D(z) = \frac{z^{-1}}{1 - 0.9z^{-1}} + \frac{\Delta}{1 - z^{-1}}, \quad (44)$$

or

$$D(z) = \frac{\Delta + (1 - 0.9\Delta)z^{-1} - z^{-2}}{1 - 1.9z^{-1} + 0.9z^{-2}}. \quad (45)$$

Application of the identification scheme yields:

$$p = q = 2$$

$$a_0 = 0.0200000000000089$$

$$a_1 = 0.9820000000000073$$

$$a_2 = -1.0000000000000422$$

for $\Delta = 0.02$. If no a priori information is available, the above data is useless. However, if the value of the bias were known, the correct parameter values could be calculated from Equation 45. On the other hand, Δ could be estimated from a_0 if the system were known to have at least one pure delay.

IV. CONCLUSION

A time-domain identification technique based on the min-max error criterion was proposed and evaluated. The results indicate that the approximation required to allow application of linear programming techniques provides an efficient solution of the identification problem and does not impair convergence or significantly affect accuracy. An iterative scheme for determining the order of a system as part of the identification procedure was proven to be effective.

Although the algorithm used in the evaluation phase is reasonably efficient, the running time and storage requirements of the computer implementation may be reduced, if necessary, by:

1. Solution of the dual problem,
2. Combining certain of the constraint equations before applying the simplex algorithm, and/or
3. Employing the revised simplex algorithm.

For a comprehensive coverage of these topics, see Hadley (1962).

APPENDIX A

THE SIMPLEX ALGORITHM

The general linear programming problem may be stated as follows: find a solution in r variables x_j to the set of m linear inequalities or equalities,

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{ir}x_r \left\{ \leq = \geq \right\} b_i, \quad (A1)$$

$$i = 1, \dots, m,$$

which maximizes a linear functional of the form

$$Z = c_1x_1 + c_2x_2 + \dots + c_rx_r. \quad (A2)$$

with

$$x_j \geq 0, \quad j = 1, \dots, r. \quad (A3)$$

All the a_{ij} , b_i , and c_j are known constants and only one of the signs \leq , $=$, \geq applies to each constraint represented in Equation A1.

The simplex algorithm is an iterative, computationally efficient method of providing a solution to the linear programming problem. The theory and development of the algorithm are described briefly in the following paragraphs. A thorough treatment of the simplex algorithm and other related linear programming topics is given by Hadley (1962).

The inequalities in the constraints (Equation A1) may each be converted to more manageable equations by the addition of suitable variables. For convenience the

constraints are rearranged with the less-than-or-equal-to constraints first, followed by the greater-than-or-equal-to and the equalities. With all $b_k \geq 0$ for reasons which will be explained shortly, the u constraints with \leq signs become

$$\sum_{j=1}^r a_{hj} x_j + x_{r+h} = b_h, \quad h = 1, \dots, u, \quad (A4a)$$

while the v constraints with \geq signs become

$$\sum_{j=1}^r a_{kj} x_j - x_{r+k} = b_k, \quad k = u+1, \dots, v+u, \quad (A4b)$$

and the equality constraints remain unchanged

$$\sum_{j=1}^r a_{pj} x_j = b_p, \quad p = u+v+1, \dots, m, \quad (A4c)$$

with all the added variables also constrained to be non-negative. If the c associated with each of these new variables is fixed at zero, solutions of the set of simultaneous linear equations developed in Equation A4 are also solutions of the original constraint equations and the problem is unchanged.

The constraints (including the non-negativity restrictions) define a closed convex set in the variable space. Any set of x_j belonging to this convex set is termed a feasible solution. Of these feasible solutions, the subset that maximizes the objective function (Equation A2)

are called optimal feasible solutions. A fundamental theorem of linear programming leading to the identification of these solutions is that, if an optimal solution exists, one or more of the extreme points of the convex set will be optimal. Furthermore, the extreme points correspond, one-for-one, to basic feasible solutions of the system of constraints, where a basic feasible solution is defined as a feasible solution with only m variables different from zero. The non-zero variables are said to be in the basis. Thus, an optimal solution may be determined by starting with an extreme point of the convex set and proceeding along "edges" to successive extreme points which increase the value of the objective function until the optimum is reached. This corresponds exactly to proceeding from one basic feasible solution to another with only one variable entering and one leaving the basis at each iteration. This is, in essence, the simplex algorithm and is illustrated in Figure A1 for the two-dimensional problem.

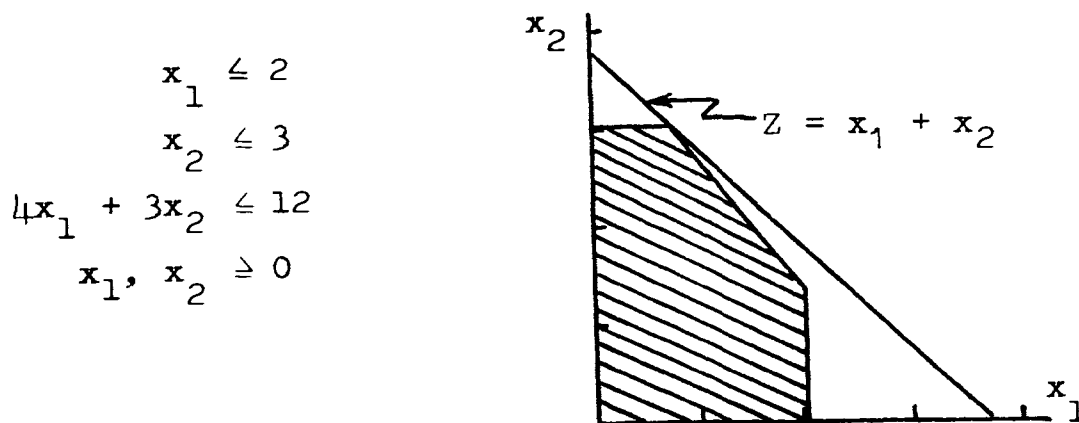


Fig. A1. The Two-Dimensional Linear Programming Problem.

Determination of an initial basic feasible solution is the starting point of the algorithm. If $b_i \geq 0$ for all i and unit vectors are present in the constraint equations corresponding to m different variables, a basic feasible solution is identified. In the more general case, artificial variables must be added to obtain an easily identifiable initial basic solution. If a large negative c_j is associated with each of these artificial variables in the objective function, they will be driven out of the solution (set to zero) in successive iterations, and thus, the formulation will revert to that of the original problem.

Before initiating the iterative procedure, the following variables are defined:

$$y_{ij} = \begin{cases} a_{ij} & \text{for } i = 1, \dots, m, \quad j = 1, \dots, r, \\ +1 & \text{for } i = 1, \dots, u, \quad j = i+r, \\ -1 & \text{for } i = u+1, \dots, v, \quad j = i+r, \\ +1 & \text{for } i = u+1, \dots, m, \quad j = i+v+u, \\ b_i & \text{for } i = 1, \dots, m, \quad j = m+v+u+1, \\ z_j - c_j & \text{for } i = m+1, \quad j = 1, \dots, m+v+u, \\ z & \text{for } i = m+1, \quad j = m+v+u+1 \end{cases} \quad (A5)$$

$$z_j - c_j = \sum_{i=1}^m c_{Bi} y_{ij} - c_j, \quad (A6)$$

$c_{Bi} = c_j$ for the i^{th} variable in the basis.

Based on these quantities, the steps in the algorithm

are:

1. Examine the $z_j - c_j$.
 - a. If all $z_j - c_j \geq 0$, the basic feasible solution is optimal.
 - b. If at least one $z_j - c_j < 0$ and has $y_{ij} > 0$ for at least one i , select the variable, x_k , with the most negative $z_j - c_j$ to enter the basis.
 - c. If for the variable selected to enter the basis all $z_j - c_j \leq 0$, the solution is unbounded.
2. Determine the variable x_r to leave the basis using

$$\frac{x_{Br}}{y_{rk}} = \min_i \left\{ \frac{x_{Bi}}{y_{ik}}, y_{ik} > 0 \right\}, \quad (A7)$$

3. Compute the new values (*) of y_{ij} by

$$y_{ij}^* = y_{ij} - y_{rj} \frac{y_{ik}}{y_{rk}}, \quad i \neq r, \quad (A8)$$

$$y_{rj}^* = \frac{y_{rj}}{y_{rk}}, \quad (A9)$$

4. Repeat all steps.

If, when an optimal solution is reached, artificial variables are present in the basis, then there is no feasible solution to the original problem.

APPENDIX B

COMPUTER PROGRAM DESCRIPTION

The computer program IDNT was generated to evaluate the performance of the identification procedure developed in this thesis. It was developed to be sufficiently general to accommodate the examples studied, and thus, it is not optimized for any particular application. Specific areas which may require modification or improvement in this respect are outlined in the Conclusion.

The organization of the executive program IDNT is illustrated in Figure B1. Basically, IDNT calculates estimates of the discrete transfer function parameters from the input data by the method presented in this thesis. The program provides the option of calculating the system impulse response from data corresponding to inputs other than an impulse. In general, IMPLZ, the subroutine which performs this function, must be generated via the techniques described on page 15; however, examples for the case of step and sinusoid inputs are presented in Appendix C. Other options include the order of the initial approximation, the bound on the error for the order determination, and the amount of data to be printed.

The following names of important variables are common to the executive routine and/or all subroutines:

F(K) - Array of input samples,
NSAM - Number of samples in input array,

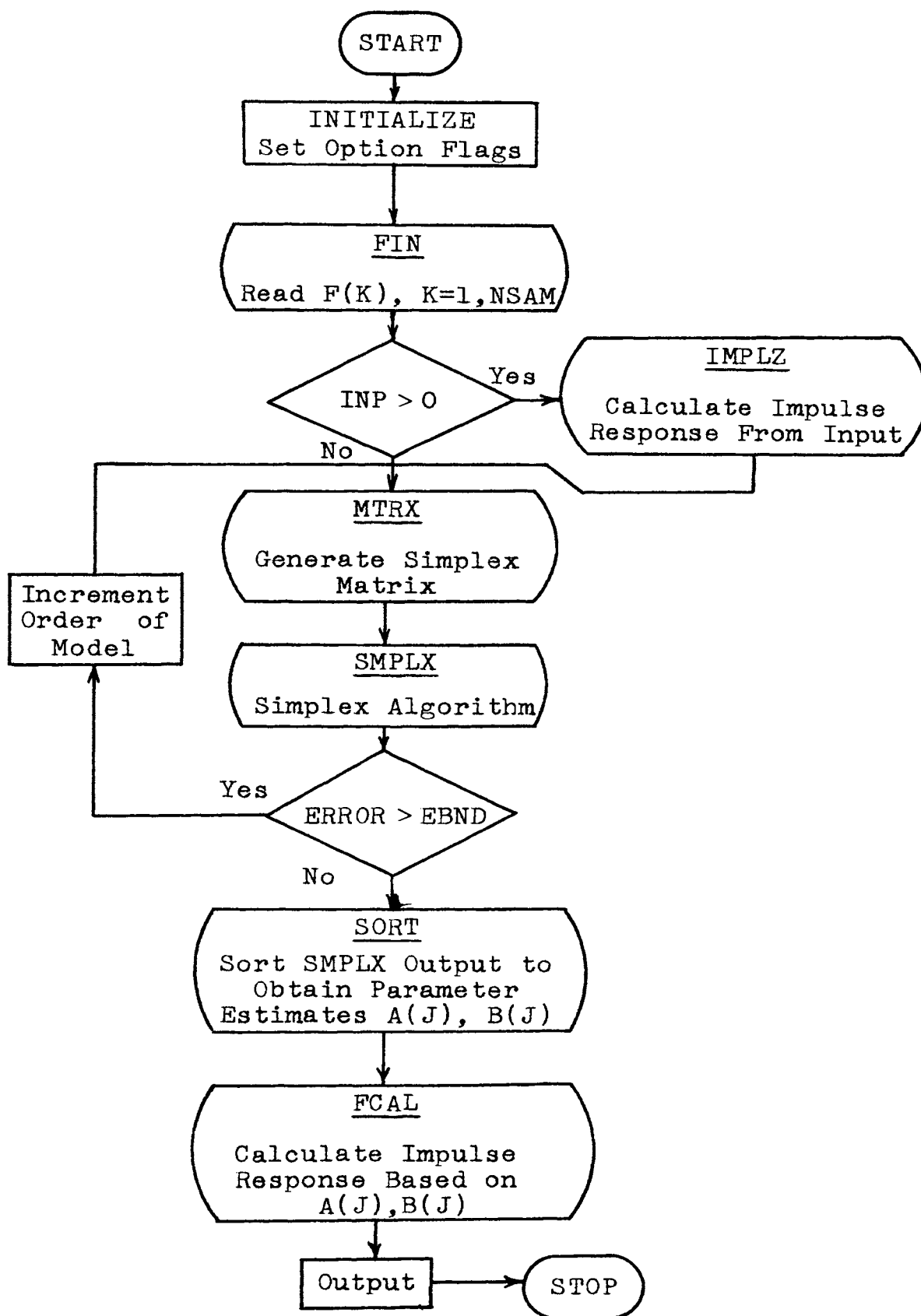


Fig. B1. Executive Program IDNT Flow Chart.

EBND - Bound on min-max error at order check point,
 INP - Flag to indicate nature of input (= 0 for
 impulse response, = 1 for response other
 than impulse),
 A(I) - Estimate of the I^{th} numerator parameter in
 the transfer function model,
 B(I) - Estimate of the I^{th} denominator parameter in
 the transfer function model,
 P - Order of the numerator,
 Q - Order of the denominator,
 Y(I,J) - Two dimensional array input to the simplex
 algorithm subroutine SMPLX (row I, column
 J),
 M1=M+1 - Total number of rows in Y matrix, and
 N1=N+1 - Total number of columns in Y matrix.

In the following, each subroutine is discussed in terms of its function and important variables.

Subroutine FIN

Subroutine FIN reads the input samples ($F(K), K=1, \dots, \text{NSAM}$) from cards in the format specified. It also reads the option variables INP, EBND, PRNT, P(initial), Q(initial), and a two-line heading. Data presented in other forms may be entered by suitable modifications of this input subroutine.

Subroutine IMPLZ

If the input samples represent the system response to an input other than an impulse, subroutine IMPLZ is called by setting INP=1 (INP=0 for normal inputs). Examples of the subroutine for step and sinusoid inputs are given in Appendix C. For other deterministic inputs, a subroutine may be developed from the techniques discussed in the section, Response of Discrete-Time Systems. The calculated impulse response is stored in the array F(K).

Subroutine MTRX

The purpose of this subroutine MTRX is to employ the techniques developed in the Discussion to transform the identification problem into a linear programming problem. The subroutine performs all manipulations required to convert the raw data into a form suitable for solution by the simplex algorithm discussed in Appendix A and outputs this formulation in the matrix Y. A flow chart of the subroutine is presented in Figure B2. The principal functions are:

1. Elimination of inequalities by addition of appropriate variables,
2. Introduction of artificial variables and formulation of an initial basic feasible solution,
3. Assignment of cost factors to each variable in the objective function,
4. Adjustment of the formulation if negative input samples are encountered,

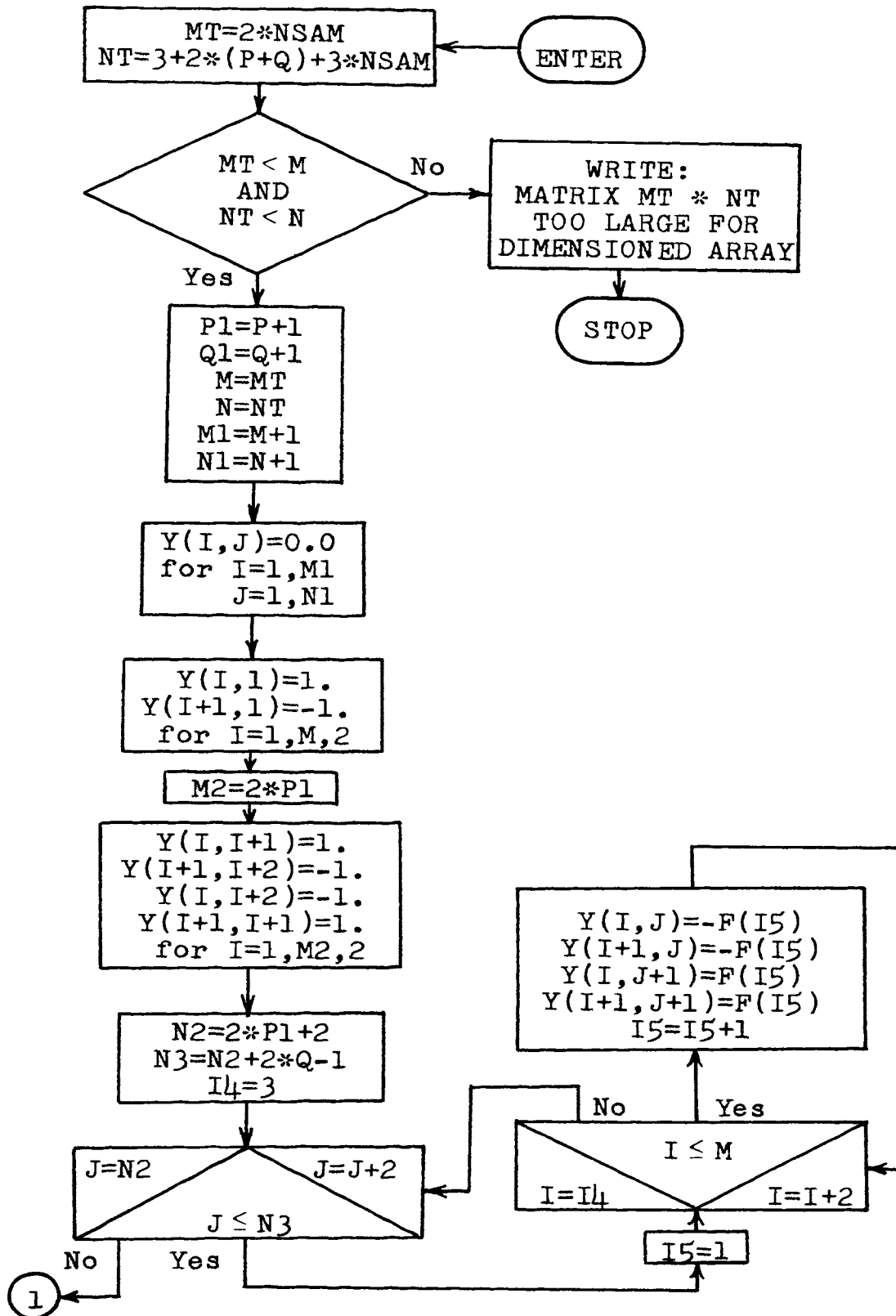


Fig. B2. Flow Chart of Matrix Generating Subroutine MTRX.

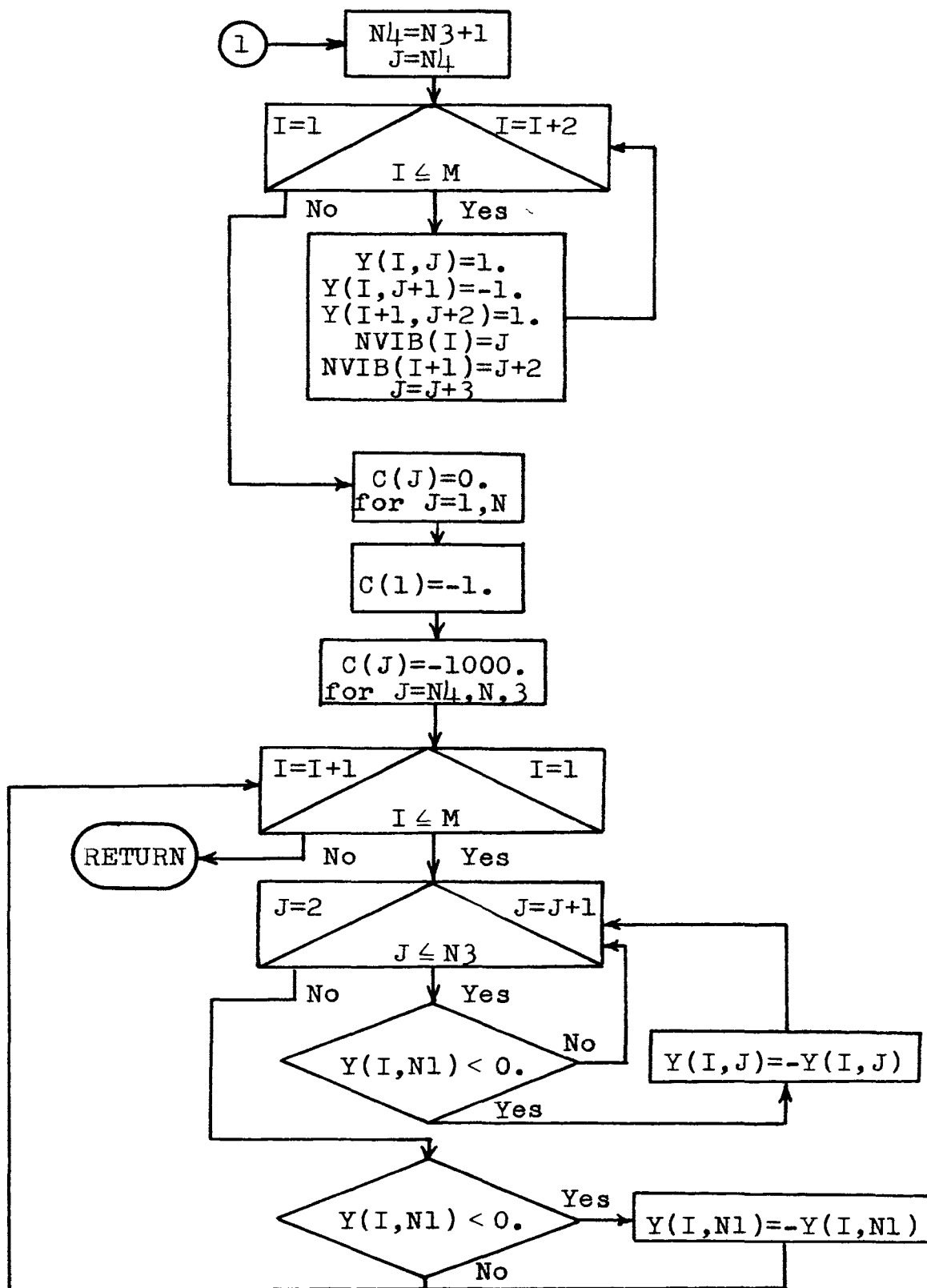


Fig. B2. (continued).

5. Introduction of complimentary variables to remove the non-negativity restraints on the parameter estimates,
6. Application of weighting factors if desireable for a particular application, and
7. Determination of the dimensions of the Y matrix and checking for field specification overflow.

The important variables not previously defined are:

C(J) - The cost factor associated with the J^{th} variable, and

NVIB(I) - A list of the variables in the basic feasible solution (in order).

Subroutine SMPLX

Subroutine SMPLX is a computer implementation of the simplex algorithm for solution of linear programming problems. It is based on the discussion presented in Appendix A and employs the notation defined in that section. To eliminate propagation of round-off errors, two minor departures from the algorithm are included in the program:

1. The $Z_j - C_j$ terms (row M1 of the Y matrix) are re-calculated at each iteration, and
2. Optimality is assumed if the $Z_j - C_j$ terms are greater than -1×10^{-14} (instead of zero).

The subroutine is flow charted in Figure B3. The important variables not previously defined are:

XMIN - the quantity described in (2) above,

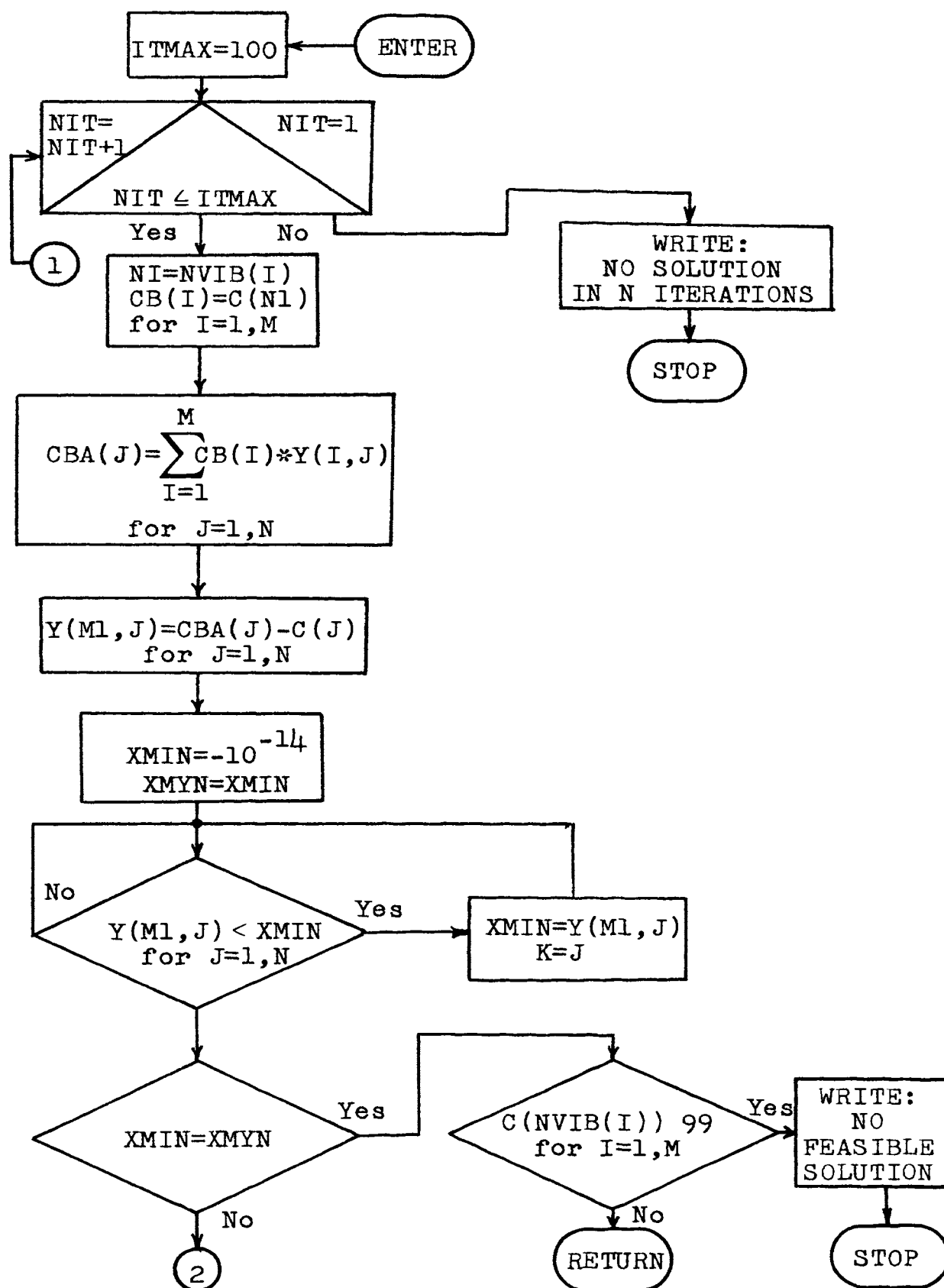


Fig. B3. Flow Chart of Simplex Algorithm Subroutine SMPLX.

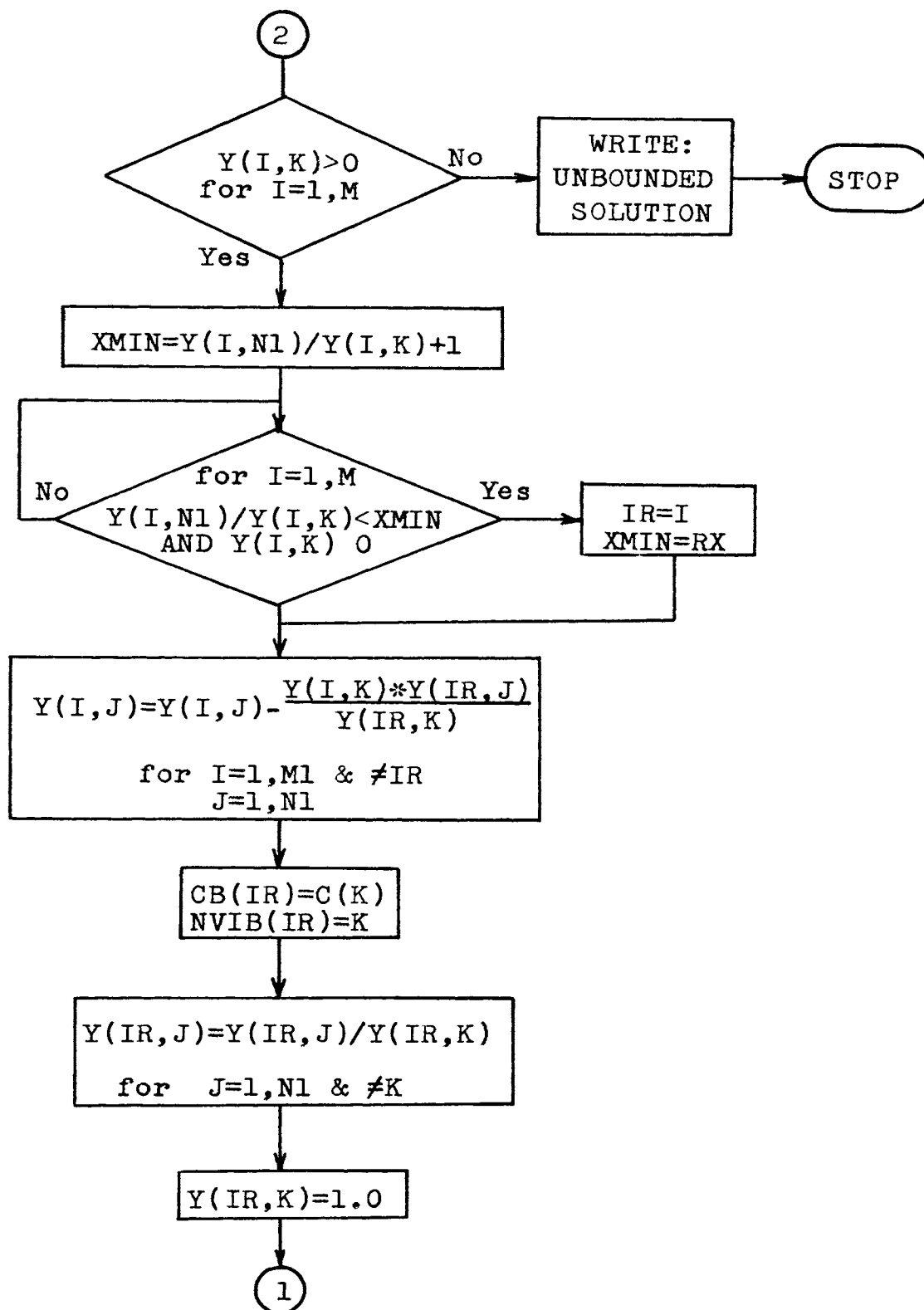


Fig. B3. (continued).

$CBA(J) - Z_j$,
 $CB(I)$ - the cost factors for the variables in the
 basic feasible solution,
 K - index of variable selected to enter the
 basis
 IR - index of variable selected to leave the
 basis, and
 $ITMAX$ - an arbitrary limit on the number of iterations.

Branch points are included for:

1. Unbounded Solution,
2. No Feasible Solution, and
3. No Solution in $ITMAX$ Iterations.

For a correctly defined problem, none of these flags should occur.

Subroutine SORT

Subroutine SORT is called, following solution of the linear programming problem by the subroutine SMPLX, to reorder or sort the variables in the solution so that the parameter estimates may be determined. The variables in the solution, $NVIB(I)$, have the values $Y(I, N1)$. The parameter estimates are calculated from the ordered solution $D(J)$. A flow chart of this subroutine is given in Figure B4.

Subroutine FCAL

Subroutine FCAL calculates the impulse sequence

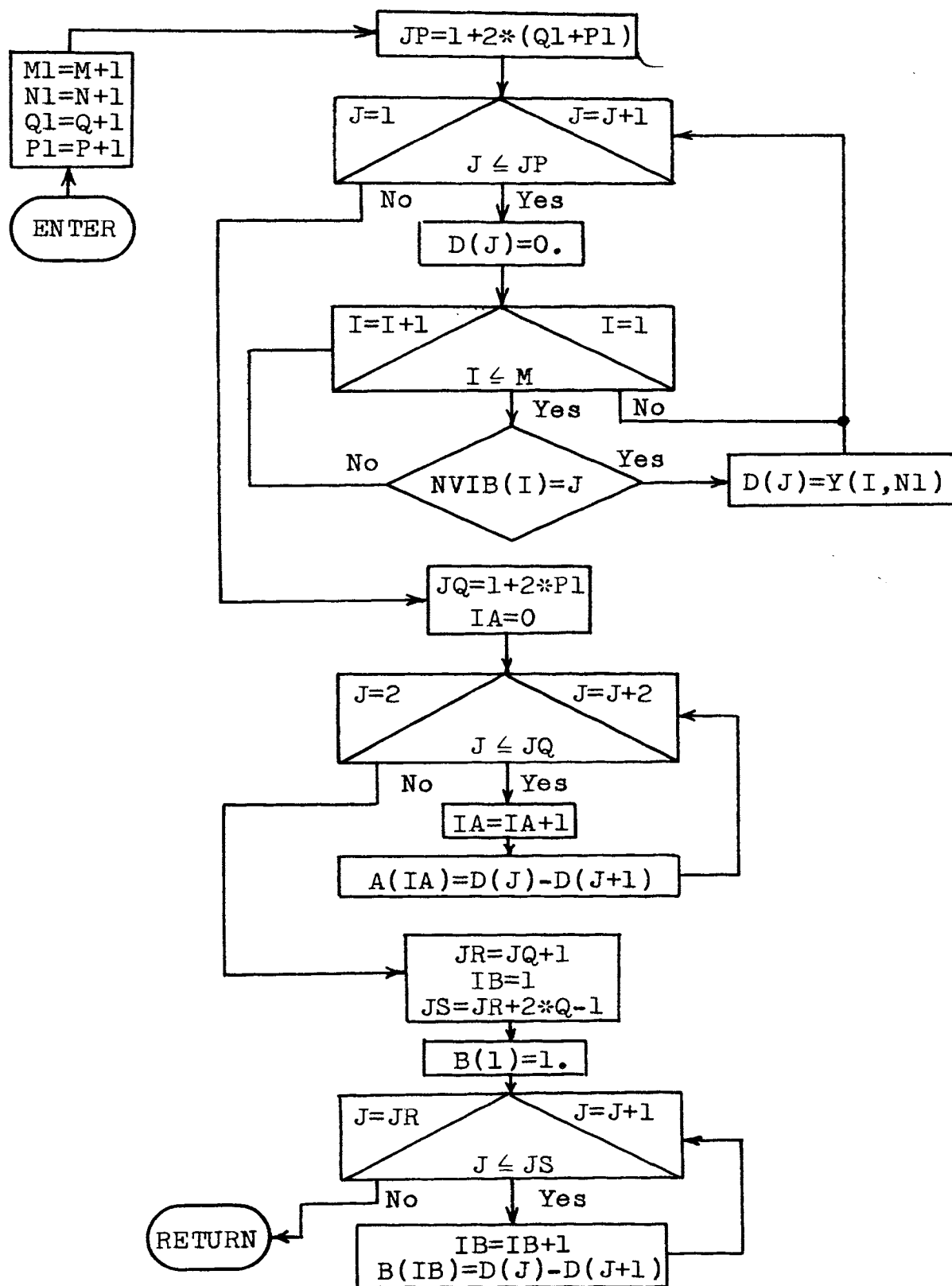


Fig. B4. Flow Chart of Sorting Subroutine SORT.

$FC(K)$, corresponding to the parameter estimates, by application of Equation 14. A flow chart is presented in Figure B5.

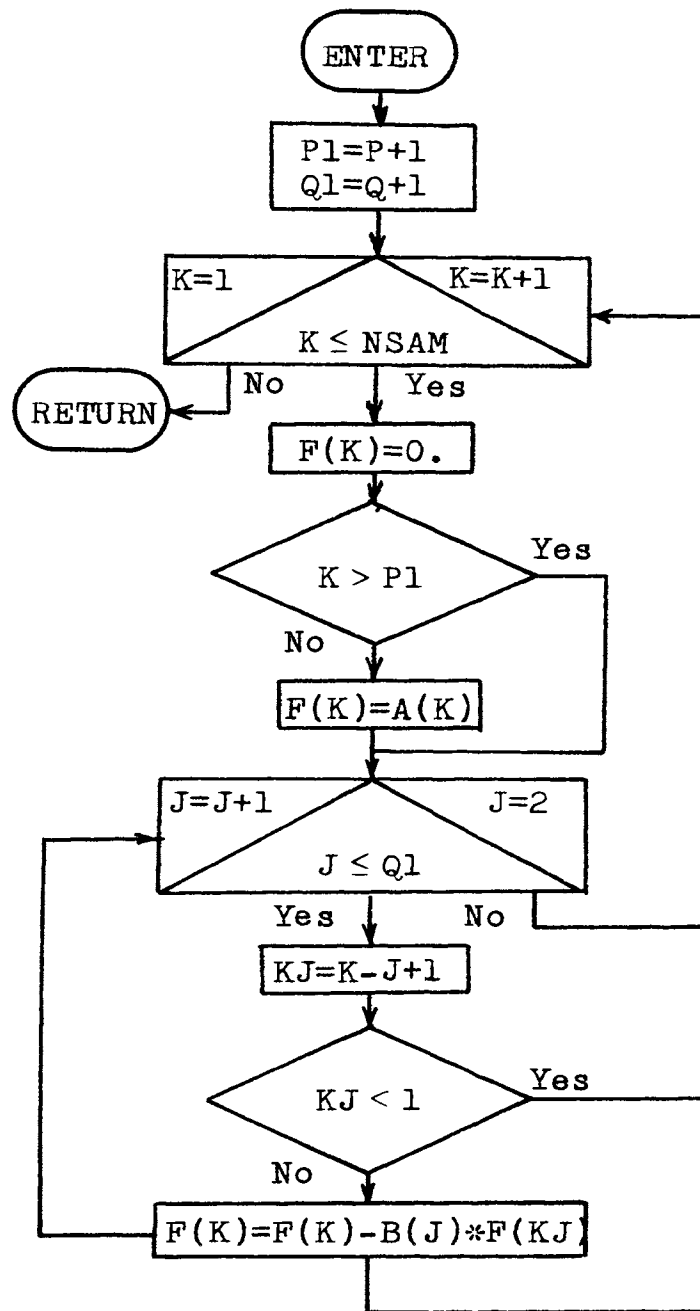


Fig. B5. Flow Chart of Z-Transform Inversion
Subroutine FCAL.

APPENDIX C

LISTING OF FORTRAN IV
COMPUTER PROGRAM IDNT

```

PROGRAM IDNT
INTEGER P,Q,P1,Q1,PI,QI
COMMON Y( 51,101),C(100),NVIB( 50),F( 20)
DIMENSION A(10),B(10),FC(20),FE(20)
WRITE (6,99)
99 FORMAT(1H1//)
CALL FIN (NSAM,P,Q,INP,EBND,PRNT)
PI=P $ QI=Q $ IF(INP.GT.0) CALL IMPLZ(NSAM)
3 CONTINUE
M=50 & N=100 & P1=P+1 & Q1=Q+1
CALL MTRX(M,N,P,Q,NSAM)
CALL SMPLX(M,N,ITMAX,NIT)
CALL SORT (M,N,P,Q,A,B)
M1=M+1 & N1=N+1
IF (ABS(Y(M1,N1)).LT.EBND) GO TO 10
P=P+1 & Q=Q+1
GO TO 3
10 CONTINUE
CALL FCAL(NSAM,P,Q,A,B,FC)
DO 30 K=1,NSAM
30 F(K)=F(K)-FC(K)
AMAX=ABS(F(1))
DO 40 K=1,NSAM
40 IF (ABS(F(K)).GT.AMAX) AMAX=ABS(F(K))
WRITE(6, 91) INP,EBND,PI,QI
91 FORMAT(/5X,*INP*,I12//5X,*EBND*,E11.1//
* 5X,*P(INITIAL)*,I5//5X,*Q(INITIAL)*,I5//)
WRITE (6, 93) (K,F(K),K=1,NSAM)
93 FORMAT(5X,*THE INPUT SEQUENCE IS*//(I7,F12.8))
WRITE(6, 92) P,Q,Y(M1,N1)
92 FORMAT(//5X,*FOR*//5X,*P = *,I7/
* 5X,*Q = *,I7/5X,*LAMDA = *,E11.4)
WRITE (6, 94) (I,A(I),B(I),I=1,P1)
94 FORMAT (///5X,*THE PARAMETER ESTIMATES ARE*//
* 6X,*I*,10X,*A(I)*,16X,*B(I)*//(I8,2E18.8//)
IF (PRNT.GT.0) WRITE (6, 95) (K,FC(K),K=1,NSAM)
95 FORMAT (///5X,*FOR THE ABOVE PARAMETER ESTIMATES*
* /5X,*THE OUTPUT SEQUENCE IS*//(I7,F12.8//)
IF(PRNT.GT.0) WRITE(6,98) AMAX
98 FORMAT (//5X,*THE ACTUAL MAXMUM ERROR IS*,E11.4/)
IF(PRNT.GT.0) WRITE(6, 97) (K,F(K),K=1,NSAM)
97 FORMAT (5X,*THE ERRORS,F(K)-FC(K), ARE*//(I7,E14.4 ))
STOP
END

```



```

      SUBROUTINE FIN(NSAM,P,Q,INP,EBND,PRNT)
      INTEGER P,Q,P1,Q1
      COMMON Y( 51,101),C(100),NVIB( 50),F( 20)
      DIMENSION A(10),B(10),HEAD(144)
C****READ HEADING AND*****
C****ALL DATA FROM CARDS*****
      READ(5,90) HEAD
      90 FORMAT(72A1)
C****2 CARD HEADING*****
      WRITE(6,90) HEAD
      READ(5,91) P,Q,INP,EBND,PRNT
      91 FORMAT(3I4,E12.2,F12.0)
      READ(5,92) NSAM,(F(K),K=1,NSAM)
      92 FORMAT(I4/( 5,F12.0))
      RETURN
      END

```

```

      SUBROUTINE IMPLZ(NSAM)
      COMMON Y( 51,101),C(100),NVIB( 50),F(20)
      DIMENSION DUM(20)
C*****
C      IMPULSE RESPONSE FROM SINE(ANT) RESPONSE
C*****
      WRITE(6,90)
      90 FORMAT('//5X,*THE SEQUENCE BELOW CORRESPONDS*/,5X,
        *      *TO A SIN(ANT) INPUT*)
      A=1.
      XCOS=COS(A)
      XSIN=SIN(A)
      DUM(1)=F(1)
      NSAM=NSAM-1
      F(1)=(F(2)-2.*XCOS*F(1))/XSIN
      DO 10 I=2,NSAM
      DUM(I)=F(I)
      10 F(I)=(F(I+1)-2.*XCOS*F(I)+DUM(I-1))/XSIN
      RETURN
      END

```

```

      SUBROUTINE IMPLZ(NSAM)
      COMMON Y( 51,101),C(100),NVIB( 50),F( 20)
C*****
C      IMPULSE RESPONSE FROM STEP RESPONSE
C*****
      F(1)=F(1)
      DO 10 I=2,NSAM
      10 F(I)=F(I)-F(I-1)
      RETURN
      END

```

```

SUBROUTINE MTRX (M,N,P,Q,NSAM)
  INTEGER P,Q,P1,Q1
  COMMON Y( 51,101),C(100),NVIB( 50),F(20)
  MT=2*NSAM
  NT=3(1+NSAM)+2(P+Q)
  IF(MT.LT.M.AND.NT.LT.N) GO TO 5
  WRITE (6,90) MT,NT
90 FORMAT (1H1//7H MATRIX,I4,2H X,I4//
  *      32H TOO LARGE FOR DIMENSIONED ARRAY)
  STOP
5  P1=P+1
   Q1=Q+1
   M=MT $ N=NT $ M1=M+1 $ N1=N+1
   DO 10 I=1,M1
   DO 10 J=1,N1
10  Y(I,J)=0
   DO 11 I=1,M,2
   Y(I,1)=1.
11  Y(I+1,1)=-1.
   I2=0
   DO 12 I=1,M,2
   I2=I2+1
   Y(I,N1)=F(I2)
12  Y(I+1,N1)=F(I2)
   M2=2*P1
   I3=0
   DO 13 I=1,M2,2
   I3=I3+2
   Y(I,I3)=1.
   Y(I+1,I3+1)=-1.
   Y(I,I3+1)=-1.
13  Y(I+1,I3)=1.
   N2=2*P1+2
   N3=N2+2*Q-1
   I4=3
   DO 15 J=N2,N3,2
   I5=1
   DO 14 I=I4,M,2
   Y(I,J)=-F(I5)
   Y(I+1,J)=-F(I5)
   Y(I,J+1)=F(I5)
   Y(I+1,J+1)=F(I5)
14  I5=I5+1
15  I4=I4+2
   N4=N3+1
   J=N4
   DO 16 I=1,M,2
   Y(I,J)=1.
   Y(I,J+1)=-1.
   Y(I+1,J+2)=1.
   NVIB (I)=1.
   NVIB(I+1)=J+2

```

```
16 J=J+3
   DO 17 J=1,N
17 C(J)=0.
   C(1)=-1.
   DO 18 J=N4,N,3
18 C(J)=-1000.
   DO 20 I=1,M
   DO 19 J=2,N3
19 IF (Y(I,N1).LT.0.) Y(I,J)=-Y(I,J)
20 IF (Y(I,N1).LT.0.) Y(I,N1)=-Y(I,N1)
   RETURN
   END
```

```

SUBROUTINE SMPLX(M,N,ITAX,NIT)
COMMON Y( 51,101),C(100),NVIB( 50),F( 20)
DIMENSION CBA(100),CB( 50)
ITMAX=100
M1=M+1
N1=N+1
DO 21 NIT=1,ITMAX
DO 10 I=1,M
NI=NVIB(I)
10 CB(I)=C(NI)
DO 11 J=1,N
CBA (J)=0.0
DO 11 I=1,M
11 CBA(J)=CBA(J)+CB(I)*Y(I,J)
DO 12 J=1,N
12 Y(M1,J)=CBA(J)-C(J)
Y(M1,N1)=0.0
DO 13 I=1,M
13 Y(M1,N1)=CB(I)*Y(I,N1)+Y(M1,N1)
XMIN=-1.E-14
XMYN=XMIN
DO 14 J=1,N
IF (Y(M1,J).LT.XMIN) K=J
14 IF (Y(M1,J).LT.XMIN) XMIN=Y(M1,J)
IF (XMIN.EQ.XMYN) GO TO 30
DO 15 I=1,M1
IF (I.GT.M) WRITE (6,90)
90 FORMAT (1H //19H UNBOUNDED SOLUTION)
IF (I.GT.M) RETURN
IF (Y(I,K).GT.0.0) XMIN=Y(I,N1)/Y(I,K)+1.0
IF (Y(I,K).GT.0.0) GO TO 16
15 CONTINUE
16 CONTINUE
DO 17 I=1,M
RX=Y(I,N1)/Y(I,K)
IF (RX.LT.XMIN.AND.Y(I,K).GT.0.0) IK=I
17 IF (RX.LT.XMIN.AND.Y(I,K).GT.0.0) XMIN=RX
DO 19 I=1,M1
IF (I.EQ.IR) GO TO 19
ALPHA=Y(I,K)/Y(IR,K)
DO 18 J=1,N1
18 Y(I,J)=(I,J)-ALPHA*Y(IR,J)
19 CONTINUE
CB(IR)=C(K)
NVIB(IR)=K
DO 20 J=1,N1
IF (J.EQ.K) GO TO 20
Y(IR,J)=Y(IR,J)/Y(IR,K)
20 CONTINUE
Y(IR,K)=1.0
21 CONTINUE
WRITE (6,91) ITMAX

```

```
91 FORMAT (1H //15H NO SOLUTION IN,I6,11H ITERATIONS)
RETURN
30 DO 31 I=1,M
    INB=NVIB(I)
31 IF (C(INB).GT. 99.) WRITE (6,92)
92 FORMAT (///5X,20HNO FEASIBLE SOLUTION)
RETURN
END
```

```

SUBROUTINE SORT (M,N,P,Q,A,B)
INTEGER P,Q,P1,Q1
COMMON Y( 51,101),C(100),NVIB( 50),F( 20)
DIMENSION D(50),A(10),B(10)
M1=M+1    $   N1=N+1
P1=P+1    $   Q1=Q+1
JP=1+2*Q1+2*P1
DO 10 J=1,JP
D(J)=0.
DO 5 I=1,M
IF(NVIB(I).EQ.J) D(J)=Y(I,N1)
IF(NVIB(I).EQ.J) GO TO 10
5 CONTINUE
10 CONTINUE
JQ=1+2*P1
IA=0
DO 15 J=2,JQ,2
IA=IA+1
15 A(IA)=D(J)-D(J+1)
JR=JQ+1
JS=JR+2*Q-1
B(1)=1.
IB=1
DO 20 J=JR,JS,2
IB=IB+1
20 B(IB)=D(J)-D(J+1)
RETURN
END

```

```
SUBROUTINE FCAL(NSAM,P,Q,A,B,F)
INTEGER P,Q,P1,Q1
DIMENSION A(10),B(10),F(20)
P1=P+1
Q1=Q+1
DO 15 K=1,NSAM
F(K)=0.
IF (K.GT.P1) GO TO 5
F(K)=A(K)
5 CONTINUE
DO 10 J=2,Q1
KJ=K-J+1
IF (KJ.LT.1) GO TO 15
10 F(K)=F(K)-B(J)*F(KJ)
15 CONTINUE
RETURN
END
```


VI. BIBLIOGRAPHY

- Anderson, G. W. et al. (1958), "A Self-Adjusting System for Optimum Dynamic Performance," 1958 IRE Nat. Conv. Rec., pt. 4, pp. 182-190.
- Bertram, J. E. (1956), "Digital Controllers for Sampled Data Systems," Trans. AIEE, vol. 75, pt. 2, pp. 151-159.
- Bigelow, S. C. and H. Ruge (1961), "An Adaptive System Using Periodic Estimation of the Pulse Transfer Function," 1961 IRE Nat. Conv. Rec., pt. 4, pp. 25-38.
- Blum, E. (1962), "A Modification of the Runge-Kutta Fourth-Order Method," Mathematics of Computation, April, pp. 176-187.
- Boxer, R. and S. Thaler (1956), "A Simplified Method of Solving Linear and Nonlinear Systems," Proc. IRE, vol. 44, January, pp. 89-101.
- Cosgriff, R. L. and R. A. Emerling (1958), "Optimizing Control Systems," AIEE Trans. (Applications and Industry), vol. 77, pt. II, March, pp. 13-16.
- Desoer, C. A. and J. Wing (1961), "A Minimal Time Discrete System," IRE Proc. on Automatic Control, AC-6, May, pp. 111-134.
- Ellington, J. P. and H. McCallion (1958), "The Determination of Control System Characteristics From a Transient Response," Proc. IEE, pt. C, vol. 105, September, pp. 370-373.
- Eykhoff, P. (1960), "Adaptive and Optimalizing Control Systems," IRE Trans. on Automatic Control, vol. AC-5, no. 2, June, p. 148.
- Freeman, H. (1965), Discrete-Time Systems, New York: John Wiley and Sons.
- Goodman, T. P. and J. B. Reswick (1955), "Determination of System Characteristics From Normal Operating Records," Trans. ASME, vol. 77, pp. 259-268.
- Gopinath, B. (1968), "On the Identification and Control of Linear Systems," Ph. D. Dissertation, Dept. of Electrical Engineering, Stanford University, California.

- Hadley, G. (1962), Linear Programming, Reading, Mass.: Addison-Wesley Publishing Co.
- Hamming, E. (1962), Numerical Methods for Scientists and Engineers, New York: McGraw-Hill.
- Ho, B. L and R. E. Kalman (1965), "Effective Construction of Linear State-Variable Models from Input/Output Data," Proc. of the Third Allerton Conf. on Circuit and System Theory, pp. 449-459.
- Joseph, P., J. Lewis, and J. Tou (1961), "Plant Identification in the Presence of Disturbances and Applications to Digital Adaptive Systems," Trans. AIEE, vol. 80 (Applications and Industry), March, pp. 18-24.
- Kalman, R. E. (1958), "Design on a Self-Optimizing Control System," Trans. ASME, vol. 80, February, pp. 468-478.
- Kelley, J. E. (1958), "An Application of Linear Programming to Curve Fitting," J. Soc. Indust. Appli. Math., vol. 6, No. 1, March.
- Kerr, R. B. and W. H. Surber, Jr. (1961), "Precision of Impulse Response Identification Based on Short, Normal Operating Records," IRE Trans. on Automatic Control, vol. AC-6, May, pp. 173-182.
- Kushner, H. (1962), "A Simple Iterative Procedure for the Identification of Unknown Parameters on a Linear Time-Varying Discrete System," Proc. of 1962 JACC, June 27-29, pp. 1-8.
- Lendaris, G. C. (1962), "The Identification of Linear Systems," Trans. AIEE (Applications and Industry), pt. 2, vol. 81, September, pp. 231-242.
- Levin, M. J. (1960), "Optimum Estimation of Impulse Response in the Presence of Noise," IRE Trans. on Circuit Theory, vol. CT-7, March, pp. 50-56.
- Levin, M. J. (1964), "Estimation of a System Pulse Transfer Function in the Presence of Noise," IEEE Trans. on Automatic Control, vol. AC-9, July, pp. 229-235.
- Margolis, S. G. (1955), "Measurement of Industrial Process Behavior," Quartely Progress Rept., Research Lab. of Electronics, Massachusetts Institute of Technology, Cambridge, Mass., April, pp. 42-46.
- Mishkin, E. and L. Braun, Eds. (1961), Adaptive Control Systems, New York: McGraw-Hill.

- McRuer, D. T. et al. (1965), "Human Pilot Dynamics in Compensatory Systems," Technical Report No. AFFKL TR-65-15, July.
- Nagumo, J. and A. Noda (1967), "A Learning Method for System Identification," IEEE Trans. on Automatic Control, vol. AC-12, June, pp. 282-287.
- Pazdera, J. S., Jr. and H. J. Pottinger, (1969), "1969 JAC Conference of the American Automatic Control Council, University of California," August 5-7, pp. 795-801.
- Price, R. and P. E. Green (1958), "A Communication Technique for Multipath Channels," Proc. IRE, vol. 46, March, pp. 555-570.
- Sage, A. and S. L. Smith (1966), "Real-Time Digital Simulation for Systems Control," Proc. IEEE, vol. 54, no. 12, December, pp. 1802-1812.
- Shinbrot, M. (1962), "A Description and Comparison of Certain Non-Linear Curve Fitting Techniques with Applications to Analysis of Transient Response Data," National Advisory Commission for Aeronautics Tech., Note No. 2622.
- Steiglitz, K. and L. E. McBride (1965), "A Technique for the Identification of Linear Systems," IEEE Trans. on Automatic Control, vol. AC-10, October, pp. 461-464.
- Tustin, A. (1947), "A New Method of Analyzing the Behavior of Linear Systems in Terms of Time Series," Journal IEE, vol. 94, May, pp. 44-53.
- Voronova, L. I. and Yu. V. Krementulo (1962), "New Method for Determining the Characteristics of Complex Dynamic Systems," NASA report TTF-11220.
- Weygandt, C. N. and N. N. Puri (1966), "Identification of a Linear System from Discrete Values of Input-Output Data," Proc. Fourth Annual Allerton Conf. on Circuit and System Theory, October 5-7, pp. 903-912.
- Wiesner, J. B. and Y. W. Lee (1950), "Experimental Determination of System Functions by the Method of Correlation," presented at IRE National Convention, New York, March.

Zaborszky, J. and R. L. Berger (1962), "An Integral Square Self-Optimizing Adaptive Control," Trans. AIEE, vol. 81 (Applications and Industry), November, pp. 256-262.

VII. VITA

Richard Loyd Warren was born January 22, 1943, in Memphis, Tennessee where he also received his primary education. In 1961, he graduated from Greenville High School, Greenville, Mississippi. Undergraduate work followed at Mississippi State University from which he received the Degree of Bachelor of Science in Electrical Engineering in May 1966.

In June 1966, he joined the Communication and Signal Processing Group within the Advanced Electronics Department of McDonnell Douglas Corporation, St. Louis, Missouri. His assignments have included analysis and hybrid simulation of advanced signal processing techniques for guidance and range sensing systems.

Professional activities include membership in Tau Beta Pi, Eta Kappa Nu, and IEEE.